Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently amended) A method of managing different types of events in a distributed computing system, having an event engine, including the steps of:

   i. providing one or more intelligent agents for receiving an event and converting the event into a standard format, and inputting [[an]] the event into the engine;

   ii. the engine extracting a rule to be applied to the event from a rules database wherein identification information within the rule identifies the event;

   iii. the engine holding the event for ~~the~~ an expiration of a specified interval;

   iv. before the expiration of the specified interval, receiving a further event from an intelligent agent, converting the further event into a standard format and inputting [[a]] the further event into the engine;

   v. the engine identifying the further event using identification information within the rule;

   vi. the engine creating and outputting a new event;

   vii. inputting the new event into the engine; and

   viii. the engine extracting a second rule to be applied to the new event from a rules database wherein identification information within the second rule identifies the new event.

2. (Currently amended) A method as claimed in claim 1 wherein the event and the further event originate from any of ~~the~~ a set of a network, an application[[,]] or an operating system residing on the distributed computing system, and hardware.

3. (Cancelled)

4. (Cancelled)

5. (Currently amended) A method as claimed in claim [[4]] 1, wherein the identification information includes:

i. an attribute;

ii. an operator; and

iii. a value.

6. (Original) A method as claimed in claim 5 wherein the specified interval is time.

7. (Currently amended) A method of managing different types of events in a distributed computing system, having an event engine, including the steps of:

i. providing one or more intelligent agents configured for receiving an event and converting the event into a standard format, and inputting [[an]] the event into the engine;

ii. the engine extracting a rule to be applied to the event from a rules database wherein identification information within the rule identifies the event;

iii. the engine creating and outputting a new event;

iv. inputting the new event into the engine;

v. the engine extracting a second rule to be applied to the new event from the rules database wherein identification information within the second rule identifies the new event;

vi. the engine holding the new event for the expiration of a specified interval;

vii. before the expiration of the specified interval, receiving a further event from an intelligent agent, converting the further event into a standard format and inputting [[a]] the further event into the engine;

viii. the engine identifying the further event using identification information within the second rule; and

ix. the engine creating and outputting a further new event.

8. (Currently amended) A method as claimed in claim 7 wherein the event and the further event originate from any of ~~the~~ a set of a network, an application, an operating system, and hardware.

9. (Cancelled)

10. (Cancelled)

11. (Currently amended) A method as claimed in claim [[10]] 8, wherein the identification information includes:

i. an attribute;

ii. an operator; and

iii. a value.

12. (Original) A method as claimed in claim 11 wherein the specified interval is time.

13. (Currently amended) A method [[a]] as claimed in claim 12 wherein the ~~outputted~~ further event is received by a user console.

14. (Currently amended) A method of managing different types of events in a distributed computing system using a management server, having an event engine, including the steps of:

i. providing one or more intelligent agents for receiving an event and converting the event into a standard format, and inputting an event into the engine;

ii. the engine extracting a first rule to be applied to the event from a rules database wherein identification information within the first rule identifies the event;

iii. the engine holding the event for ~~the~~ an expiration of a specified interval;

iv. before the expiration of the specified interval, receiving a further event from an intelligent agent, converting the further event into a standard format and inputting [[a]] the further event into the event engine;

v. the engine extracting a second rule, to be applied to the further event from the rules database wherein identification information within the second rule identifies the further event;

vi. the engine creating and outputting a new event;

vii. before the expiration of the specified interval, inputting the new event into the engine;

viii. the engine identifying the new event using identification information within the first rule; and

ix. the engine creating and outputting a further new event.

15. (Original) A method as claimed in claim 14 wherein the event and the further event originate from any of the set of a network, an application, an operating system, and hardware.

16. (Cancelled)

17. (Cancelled)

18. (Currently amended) A method as claimed in claim [[17]] 15, wherein the identification information includes:

i. an attribute;

ii. an operator; and

iii. a value.

19. (Original) A method as claimed in claim 18 wherein the specified interval is time.

20. (Currently amended) A method a claimed in claim 19 wherein the ~~outputted~~ further event is received by a user console.

21. (Currently amended) A method of managing different types of events in a distributed computing system including the steps of:

i. receiving an event and converting the event into a standard format;

ii. extracting a rule to be applied to the event from a rules database wherein identification information within the rule identifies the event;

iii. when specified within the rule performing one of:

a) creating a new event; or

b) holding the event;

wherein during the method at least one rule specifies performance of step (a) and at least one rule specifies performance of step (b); and

iv. repeating steps (i) to (iii) at least once;

wherein at least one received event in step (i) is a new event created in step (iii) (a).

22. (Cancelled)

23. (Currently amended) A method of managing different types of events in a distributed computing system including the steps of:

i. processing an event by:

a) receiving the event and converting the event into standard format;

b) extracting one or more rules which match the event from a rules database;

c) discarding the event if at least one of the rules specifies that the event is to be discarded;

d) holding the event if at least one of the rules specifies that the event is to be held for a period of time;

e) altering the event or creating a new event if at least one of the rules specifies that the event is to be altered or a new event created; and

f) outputting the event if all rules specify that the event is to be outputted;

wherein if the event is discarded then neither of steps (d) and (e) will proceed;

ii. holding the event for the longest period of time specified by the rules if the event is specified to be held; and

iii. repeating step (i) if the event was held in step (ii).

24. (Currently amended) A system for managing different types of events in a distributed computing system including:

i. a plurality of event agents adapted to receive data from a source, to create an event from the data, convert the event into a standard format and to transmit the event to a central event system; and

ii. a central event system including:

a) a rules database adapted to store a plurality of rules, each rule including:

I. identification information specifying to which events the rule relates; and

II. an action wherein the action is one of outputting the event, discarding the event, holding the event, or creating a new event;

wherein, where the action is holding the event the rule further includes:

I. a condition; and

II. a further action wherein the further action is one of outputting the event, discarding the event, holding the event, creating a new event, or creating a new event and transmitting the new event back into the processing engine; and

b) a processing engine adapted to receive events, to extract rules from the rules database, to identify which rules apply to the events using the identification information within the rule, to perform the action specified within the applicable rules, and to perform the further action specified within the applicable rules when the corresponding condition is satisfied.

25. (Original) A system as claimed in claim 24 including one or more user consoles adapted to receive one or more of the events outputted by the central event system.

26. (Currently amended) A system as claimed in claim 25 wherein the source is any one of ~~the~~ a set of a database, an application, an operating system, and hardware.

27. (Original) A system as claimed in claim 26 wherein the identification information includes:

i. an attribute;

ii. an operator; and

iii. a value.

28. (Currently amended) A computer system for ~~effecting~~ executing the method of claim 21.

29. (Cancelled)

30. (Currently amended) Storage media containing software for executing the method of claim 21. ~~as claimed in claim 29.~~

31. (Cancelled)

32. (Currently amended) Storage media containing software for executing the system of claim 24. ~~as claimed in claim 31.~~